

Tinker-OpenMM: Absolute and Relative Alchemical Free Energies using AMOEBA on GPUs

Matthew Harger ^[a], Daniel Li,^[a] Zhi Wang,^[b] Kevin Dalby,^[c] Louis Lagardère,^[d] Jean-Philip Piquemal ^[e,f], Jay Ponder,^[b] and Pengyu Ren^{*[a]}

The capabilities of the polarizable force fields for alchemical free energy calculations have been limited by the high computational cost and complexity of the underlying potential energy functions. In this work, we present a GPU-based general alchemical free energy simulation platform for polarizable potential AMOEBA. Tinker-OpenMM, the OpenMM implementation of the AMOEBA simulation engine has been modified to enable both absolute and relative alchemical simulations on GPUs, which leads to a ~ 200 -fold improvement in simulation speed over a single CPU core. We show that free energy values calculated using this platform agree with the results of Tinker simulations for the hydration of organic compounds

and binding of host-guest systems within the statistical errors. In addition to absolute binding, we designed a relative alchemical approach for computing relative binding affinities of ligands to the same host, where a special path was applied to avoid numerical instability due to polarization between the different ligands that bind to the same site. This scheme is general and does not require ligands to have similar scaffolds. We show that relative hydration and binding free energy calculated using this approach match those computed from the absolute free energy approach. © 2017 Wiley Periodicals, Inc.

DOI: 10.1002/jcc.24853

Introduction

Free energy is the driving force for spontaneous molecular processes and accurate alchemical free energy calculations can benefit a broad range of chemical and biomedical applications.^[1–5] The accurate prediction of the binding affinities for ligands to their target proteins has been a great challenge in the computational drug development process.^[6] Today, it is common to utilize empirical docking algorithms in the identification of potential lead compounds.^[7–11] However, to screen large ligand libraries in a short amount of time, empirical docking typically relies on incomplete physics models,^[12] and only account for limited system dynamics (such as loop flexibility) when predicting ligand affinity.^[13] These limitations result in a lack of the accuracy necessary for lead optimization.^[14,15] The calculation of ligand binding free energies from elaborated molecular simulations has also been limited by a combination of underlying force fields and sampling algorithms.^[16,17]

One pathway for the calculation of binding free energies is the double decoupling approach. In this approach, one includes a parameter (λ) that controls the interaction of a ligand with its environment. When transitioning from $\lambda = 1$ (full ligand intermolecular interaction) to $\lambda = 0$ (no ligand intermolecular interaction), a ligand's interaction with its environment is evaluated. Simulations of the system are conducted with the solvated ligand and the protein-ligand complex, and the binding free energy is calculated as the complexation energy minus the solvation energy, plus standard state and other corrections.^[18] In this methodology restraints^[19] are often used to keep the ligand bound to the protein complex throughout the decoupling process. The

magnitude of this restraint term is then analytically corrected for.

Another major class of approaches of binding free energy involve the calculation of the potential of mean force. In these approaches, pioneered by the Roux lab,^[20] one calculates the average force needed to maintain a system in a given configuration (e.g., the distance and orientation between a ligand and the active site). Free energy is then calculated by calculating

[a] M. Harger, D. Li, P. Ren

Department of Biomedical Engineering, University of Texas at Austin, Austin, Texas 78712

E-mail: pren@mail.utexas.edu

[b] Z. Wang, J. Ponder

Department of Chemistry Washington University in St. Louis, St. Louis, Missouri 63130

[c] K. Dalby

Division of Chemical Biology and Medicinal Chemistry, University of Texas at Austin, Austin, Texas 78712

[d] L. Lagardère

Institut des Sciences du Calcul et des Données, UPMC Université Paris 06, F-75005, Paris, France

[e] J.-P. Piquemal

Laboratoire de Chimie Théorique, Sorbonne Universités, UPMC, UMR7616 CNRS, Paris, France

[f] J.-P. Piquemal

Institut Universitaire de France, Paris Cedex 05, 75231, France

Contract grant sponsor: National Institutes of Health; Contract grant numbers: R01GM106137 and R01GM114237; Contract grant sponsor: CPRIT; Contract grant number: RP160657; Contract grant sponsor: Robert A. Welch Foundation; Contract grant number: F-1691; Contract grant sponsor: National Science Foundation; Contract grant number: DGE-1610403; Contract grant sponsor: French CNRS through a PICS (UPMC and UT Austin is acknowledged); Contract grant sponsor: French state funds (CalSimLab and the ANR within the Investissements d'Avenir program); Contract grant number: ANR-11-IDEX-0004-02

© 2017 Wiley Periodicals, Inc.

the work integral from the starting to ending distances. To obtain energy data on all relevant distances, a biasing process such as steered MD^[21,22] or umbrella sampling^[20,23] is often used. The advantage of this technique is that it allows for the collection of free energy profiles, including information about the energy barriers to binding. The main challenge of this approach is the difficulty in defining an appropriate reaction coordinate for the biasing process. Therefore, this technique has been mostly applied to systems such as channel proteins^[24,25] that have an obvious pulling dimension. However, this technique can also be applied to general protein–ligand binding.^[26–28]

The free energy between the bound and unbound states in either approach can be sampled by using various techniques such as free energy perturbation (FEP),^[3] thermodynamic integration (TI),^[29] metadynamics,^[30–32] or orthogonal space random walk (OSRW).^[33,34] A common method for calculating the free energy between neighboring states in alchemical perturbation is the Bennett acceptance ratio (BAR).^[35] The free energy of binding can then be calculated as the difference between the ligand–host interaction energy and the ligand–water interaction energy. In thermodynamic integration, one utilizes lambda much like in setting up a simulation for BAR and calculate the numerical integration of $\langle \partial H / \partial \lambda \rangle_\lambda$ from lambda = 0 to lambda = 1.^[29] Compared to BAR, it can be difficult to determine which discrete values of lambda should be used, as convergence can be difficult in regions of high curvature of $\langle \partial H / \partial \lambda \rangle_\lambda$. Due to this, comparison studies^[36] have suggested that TI simulations may require more states than BAR to reach converged free energies. However, TI simulations require less postsimulation processing than BAR-based approaches.

The second ingredient of free energy simulations is the choice of force field. Popular force fields include CHARMM^[37–40] and AMBER.^[41–44] More recent advances have resulted in the development of force fields with more complex electrostatics models, particularly incorporation of polarization. General polarizable force fields include polarizable multipole-based AMOEBA,^[45–47] polarizable OPLS,^[48–50] fluctuating charge,^[51,52] and Drude-Oscillator.^[53–55] based CHARMM force fields. The defining feature of the AMOEBA force field we have been developing is its electrostatic model based on permanent atomic multipoles, as well as many-body polarization through induced dipoles. These added terms, while computationally expensive, allow for a more rigorous modeling of ligand–protein interaction, particularly at short range, than is possible using a fixed-charge-based force field.

Previous work using AMOEBA force field has shown an accurate recapitulation of experimental free energies in small molecules hydration,^[55,57–59] metal ion hydration,^[60–62] as well as ligand binding in synthetic hosts,^[63] and protein systems.^[47,64–68] The inclusion of a complex electrostatic force leads to increasing computational cost, so that potential it can benefit even more from parallel computing of protein-scale systems consisting of tens of thousands of atoms. Earlier implementations of AMOEBA in Tinker have utilized OpenMP,^[69] which allows for limited parallelism on commercially available CPUs. Massively parallel computation using AMOEBA is possible on supercomputers using the

Tinker-HP package.^[70,71] In addition, AMOEBA has been previously implemented in OpenMM, enabling massively parallel molecular dynamics simulations on GPUs.^[72,73] To enable alchemical free energy calculations in OpenMM, we have incorporated “lambda” into force and energy calculation via a soft-core approach,^[74] which is necessary to remove the singularities in van der Waals (vdW) interactions that occurs when atoms are in close contacts.^[75] In addition, we modified the Tinker-OpenMM interface to allow for perturbation of the electrostatic force via the scaling of electrostatic parameters. Another feature of OpenMM that is now supported by the Tinker-OpenMM interface is the addition of support for the CustomCentroidBondForce. This addition enables the coupling of a two groups of atoms (such as a ligand and its binding site).

Compared to the state of CPU alchemical free energy calculations, GPU alchemical free energy calculations is still in its infancy. It is possible to perform MD simulations on GPUs using a few software, including AMBER,^[76] NAMD,^[77] and OpenMM.^[72] However, very few GPU platforms have yet supported alchemical simulations. In addition to the work with OpenMM-AMOEBA described here, the YANK package for the use of OpenMM to simulate AMBER force fields is currently in development. Therefore, the AMOEBA on GPU implementation described here (Tinker-OpenMM) constitutes the first available platform for free energy perturbation simulations on GPUs using a polarizable force field.

It is not always necessary to compute the absolute alchemical free energy, and binding or solvation energies relative to a reference ligand are sufficient. In those cases, it may be advantageous to calculate relative energies instead of absolute energies. Many previous relative binding free energy calculation use a “dummy atom” single topology approach^[78–82] where a pair of ligands are simulated as a common core of atoms connected to a set of atoms sufficient to describe both desired molecules. This dummy atom approach has been used to calculate a number of molecular properties, including binding free energies.^[79–83] Previous work with the AMOEBA force fields on CPUs have accurately calculated the relative binding free energies of ligands to trypsin using a single topology approach.^[66,67] The weakness of this scheme is that it is not general; it is more suitable for pairs of molecules with significant chemical similarity. A different approach is that of dual topology free energy calculation, where two ligands are always present in the binding pocket. Relative complexation free energy is calculated via a path starting in a state with fully ligand 1–environmental interaction, and ending at a state of fully ligand 2–environmental interaction. Dual topology free energy calculations have been possible in CHARMM since the late 80s^[84] and have more recently been implemented in AMBER.^[76] However, this dual topology scheme is more difficult to implement in a polarizable force field due to the complexity of the electrostatics making it difficult to selectively “scale” the polarization between two ligands. By utilizing a pathway where only one ligand is charged during any perturbation step, we were able to avoid this complication.

Currently, the ability to perform GPU-based platform alchemical simulations, particularly for polarizable force fields,

has been limited. In this work, we created Tinker-OpenMM, an OpenMM implementation of AMOEBA that enables alchemical free energy calculations on GPUs, while also adding the capability to perform dual topology simulations to both the Tinker^[85] and OpenMM^[72,73] platforms. We then proceed to test the GPU-based free energy calculations for hydration free energies of aromatic systems,^[86] absolute and relative binding free energies of the sampl4 host-guest systems.^[87]

Implementation Details

Tinker-OpenMM interface

Tinker-OpenMM is built using an interface to pass tinker coordinates and parameters to OpenMM. Tinker reads in the input key and coordinate files, and passes the relevant variables in to a C++ script. This script then uses the OpenMM C API to create the relevant OpenMM parameters and forces, and initiates GPU Molecular Dynamics simulation. Coordinate saving is then managed by occasionally transferring atomic coordinates and velocities from the GPU to main system memory. Tinker then saves these outputs in Tinker coordinate and velocity files, enabling post-processing by Tinker commands (e.g., BAR). This interface was originally created by Mark Friedrichs, Lee-Ping Wang, Kailong Mao, and Chao Lu.

Absolute binding free energy

In this work, we employ double-decoupling and alchemical perturbation to compute free energy of binding. First, the electrostatic interactions between the ligand and its environment (water or protein/water) are scaled from 0 to 100% in a series of simulations. With no electrostatic interaction between ligand and surroundings, a series of simulations are run where the (softcore) vdW interactions between ligand and environment are scaled. The path utilized for absolute complexation simulations is shown in Figure 1. This process is also repeated in an aqueous environment to account for hydration free energy.

After running these simulations, the Bennett acceptance ratio (BAR) method is used to calculate the free energy difference between each pair of neighboring states. Since energy is a state function, we can calculate the total complexation energy as the sum of many small perturbations in ligand-environment interaction strengths. The binding energy is calculated as the complexation free energy, minus the hydration free energy, with the addition of several corrections explained below.

When conducting alchemical perturbation, it is necessary to denote which atoms belong to the ligand. In the simulation system, the ligand atom indices are identified by using the *ligand* keyword in the key file (e.g., "*ligand* -1 14" denotes that atoms 1 through 14 belong to a ligand).

Alteration of the electrostatic interactions between the ligand and its environment is accomplished via the scaling of the electrostatic parameters passed from the Tinker interface to OpenMM. The atomic charge, dipole, quadrupole, and polarizability of all ligand atoms are each multiplied by the

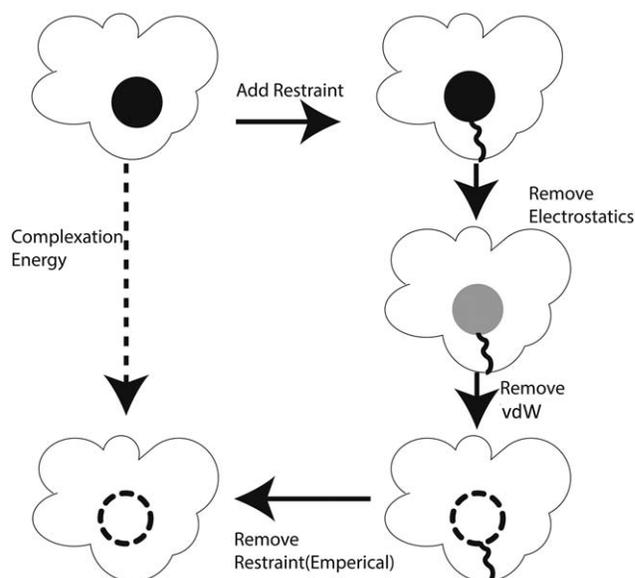


Figure 1. Thermodynamic path used to calculate the absolute complexation energy of a ligand using a double-decoupling approach.

current simulation electrostatic lambda value (between 0 and 1), which is denoted by the *ele-lambda* keyword. This results in no electrostatic interaction between the ligand and its environment when *ele-lambda* = 0, and full interaction strength when *ele-lambda* = 1. This methodology also "turns off" the intra-ligand electrostatic interactions. When calculating hydration free energy, the intra-ligand/solute electrostatic contributions are added back by "growing" the electrostatic parameters for ligand alone (in gas phase). However, when calculating binding free energy, this contribution is exactly canceled by an equal omission in the ligand-solvent interaction.

When conducting alchemical perturbation simulations, the change in energy and structure that results from each perturbation needs to be relatively small. To avoid the numerical instability of the standard vdW function when the ligand-environment interaction approaches zero, a softcore buffered 14-7 vdW (energy equation shown below) has been used to calculate the forces and energies.^[67]

$$U_{ij}^{vdw} = \lambda_{ij}^5 \epsilon_{ij} \frac{1.07^7}{0.7(1-\lambda_{ij})^2 + (\rho_{ij} + 0.07)^7} \left(\frac{1.12}{0.7 * (1-\lambda_{ij})^2 + \rho_{ij}^7 + 0.12} - 2 \right) \quad (1)$$

Here ϵ_{ij} is the well depth, and ρ_{ij} represents the current interatomic distance divided by r_{\min} , the interatomic distance that results in the lowest vdW energy. To use this softcore vdW force, we need to assign the appropriate value of the lambda parameter λ_{ij} . In this implementation, each ligand atom is assigned a lambda value equal to the *vdW-lambda* keyword value in the simulation input key file. Each nonligand atom is assigned a lambda value of 1. When calculating a pairwise vdW interaction, it is necessary to have a set of combining rules to convert two atomic vdW lambdas into a combined, λ_{ij} . For a pair of atom i and j , λ_{ij} is determined as the lesser of λ_i

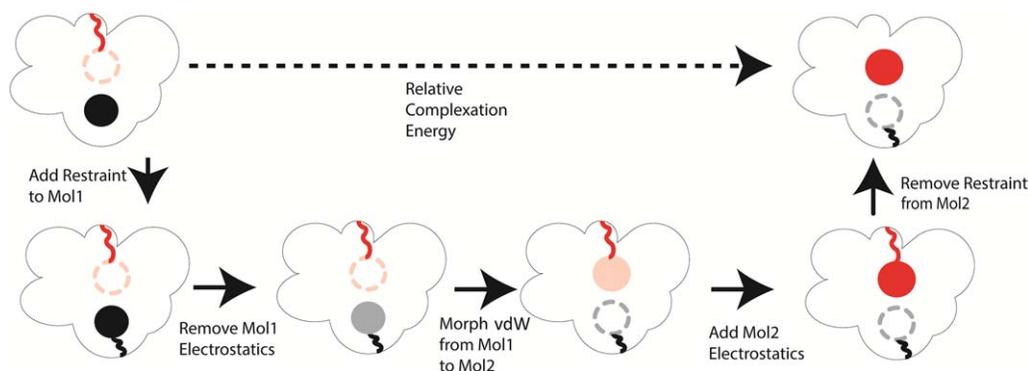


Figure 2. Path used to determine the relative complexation interaction energy of two ligands using a dual topological approach. [Color figure can be viewed at wileyonlinelibrary.com]

and λ_j . If the two lambda values are identical (as is the case in an intraligand or water–water interaction), $\lambda_{ij}=1$.

To ensure that the ligand stays in the binding pocket even when intramolecular interactions are weak, a distance restraint ($k(r-r_0)^2$) is applied between the centers of mass of the ligand and the center of the binding pocket. The bias introduced by the restraint is corrected for at the start and end of our thermodynamic path. The restraint correction at the end of simulation where no intermolecular interaction between ligand and environment is given by^[88]

$$\Delta G_{\text{restraint}} = RT \ln \left[C^0 \left(\frac{\pi RT}{k} \right)^{\frac{3}{2}} \right] \quad (2)$$

Here, C^0 represents standard state concentration (1 mol/L). In this work, we use a force constant (k) of 15 Kcal/mol/Å², and this correction amounts to 6.25 Kcal/mol.

To remove the ligand restraint from the system with full ligand–protein interaction, we repeat the simulation but with the restraint off. The free energy difference between the two simulations is then calculated using BAR. Alternatively, one could also gradually turn off the restraint while the interaction strength between ligand and protein increases so that no additional correction is needed.

Dual-topology-based relative free energy

Relative binding free energy can potentially be calculated more reliably as it avoids simulation of the nonligand bound form of the protein. In this implementation of the calculation of relative binding free energies, we take a thermodynamic path where we first reduce ligand 1's electrostatic parameters to zero magnitude. We then proceed to reduce the vdW interactions between ligand 1 and environment, while simultaneously increasing the vdW interactions between ligand 2 and environment. Finally, we increase ligand 2's electrostatic parameters from zero to full. The path we used to calculate relative complexation energy (ligand binding to receptor in water) is shown in Figure 2. Since the two ligands are never charged at the same perturbation step, ligand 1 and 2 never interact with each other (the vdW interactions are also turned

off via the soft-core formula), which requires minimal changes to the electrostatic force in the existing OpenMM code.

To run the simulations in our thermodynamic path, we require independent (*ligand 1* and *ligand 2*) keywords to denote the indices of ligand 1 and ligand 2, respectively. The electrostatic perturbation segments of our path require that we independently control the electrostatic interaction of ligand 1 and ligand 2. This is accomplished by having two electrostatic lambda keywords (*ele-lambda1* and *ele-lambda2*, respectively). The charge, dipole, quadrupole, and polarizability of each ligand is multiplied by the appropriate *ele-lambda* variable.

When perturbing the vdW force, we need to assign each ligand atom the correct lambda value. The vdW-lambda of all ligand 1 atoms is equal to the value specified by the *vdW-lambda* keyword, and vdW-lambda of all ligand 2 atoms is equal to 1 minus *vdW-lambda*. Therefore, changing the *vdw-lambda* keyword from 1.0 to 0.0 results in removing all ligand 1–environment interactions while setting all ligand 2 atoms to full vdW interaction with the environment.

When conducting relative binding simulations or BAR energy calculations, we need to ensure that the two ligands do not interact *via* the vdW force. Therefore, we need a way for our vdW force and energy calculations kernels to know which ligand each atom belongs to. This is accomplished by adding an internal variable to the vdW force used to designate which ligand (if any) an atom belongs to. This variable is equal to 0 for environmental (nonligand) atoms, 1 for ligand 1, and 2 for ligand 2. Each pairwise vdW interaction is checked to ensure that ligand 1–ligand 2 interactions are omitted.

The relative binding free energy is calculated as the relative complexation energy minus the relative hydration energy. Note that if one uses the same force constant for ligand–receptor restraint for all simulations, the restraint correction discussed above is identical for both ligands and drops out in the relative binding free energy.

Methods

Simulation setup

Prior to all simulation, the system energy was minimized to 1 Kcal/mol/Å to avoid close atomic contacts. All simulations

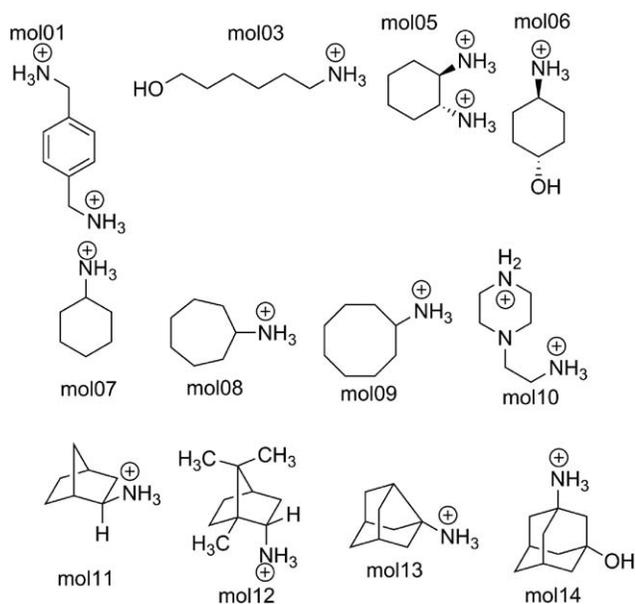


Figure 3. Structures of the 12 sampl4 molecules utilized in this study.

were run under OpenMM mixed precision mode. Ewald cutoff was set to 7.0 Å, with a 12 Å vdW cutoff in both simulations. All simulations converge the induced dipole moments between iterations to <0.00001 D. Sampl4 and aromatic simulations use a cubic box of 40 Å an Ewald grid of 48 × 48 × 48, while the larger bench7 dataset uses an Ewald grid of 64 × 64 × 64 and a cubic box of 62.23 Å. Example Tinker key files are included in the Supporting Information.

Molecular dynamics

Perturbation steps for absolute binding and solvation simulations were conducted with a stepwise reduction of the *ele-lambda* keyword, followed by a stepwise reduction of the *vdw-lambda* keyword at 0 *ele-lambda*. MD used a RESPA integrator, and a BUSSI thermostat. Information on what perturbation steps were used is included in the Supporting Information.

Relative binding and solvation simulations were conducting starting with the *ele-lambda1* and *vdw-lambda* keywords at 1.0, and the *ele-lambda2* keyword at 0.0. In a series of simulations, the *ele-lambda1* keyword is then gradually reduced to 0.0. This is followed by simulations with a stepwise reduction of *vdw-lambda1* to 0.0, then a stepwise increase of *ele-lambda2* to 1.0.

All CPU simulations were conducted using Tinker dynamic.x for 1 ns with a 2 fs time step and snapshots saved every 1 ps. Each GPU perturbation simulation was conducted using dynamic_omm for 5 ns, with a 2 fs time-step and snapshots saved every 2 ps (except for relative free energy simulations, which had snapshots saved every 1 ps). All simulations were conducted at 298 K.

Bennett acceptance ratio

Bar was computed using Tinker's BAR program. This program iterates between the two equations below until convergence:

$$e^{-\beta\Delta F} = \frac{\langle f(\beta(U_2 - U_1 - C)) \rangle_1}{\langle f(\beta(U_1 - U_2 + C)) \rangle_2} \quad (3)$$
$$C = \Delta F$$
$$f(x) = \frac{1}{1 + e^x}$$

For all CPU-based trajectories, BAR used frames 400 to 1000 for calculation, with the initial 400 ps equilibration discarded. For absolute free energy trajectories generated on the GPU, BAR used frames 1 to 2500(0–5 ns) for calculation. For the relative free energy trajectories generated on the GPU, BAR used frames 1 to 5000(0–5 ns) due to more frequently saved snapshots.

Hydration of aromatic compounds

Parameters for the aromatic molecules were previously generated.^[86] Structures of the 10 compounds are shown in Figure 3. Initial simulation systems were generated by solvating each ligand in water boxes using the Tinker commands *solvate* and *crystal*. Initial structures for relative HFE simulations were generated by concatenating ligand 2's coordinates to the solvated ligand 1 pose.

To calculate the absolute hydration free energy, it is necessary to correct for the contribution of intramolecular electrostatics as we scale the solute electrostatic parameters in “disappearing” or “growing” the solute molecule. The intrasolute electrostatic energy was calculated by running simulations on CPU (this same value was used for both the CPU and GPU simulations). Each molecule was simulated alone in a nonperiodic system at *ele-lambda* values of 0, 0.1, ... and 1.0. Simulations were run for 1 ns using a time step of 0.1 fs, with structures saved every 0.5 ps at constant volume of 40.0 Å with temperature at 298 K. The intrasolute electrostatic energy was then calculated using BAR.

Sampl4 binding simulations

Parameters and starting poses for 12 molecules of the sampl4 dataset were generated as described previously.^[63] Structures of the sampl4 ligands utilized in this study are shown in Figure 4. Relative binding poses were generated as in the relative aromatic simulations.

The final absolute binding energy was calculated as ΔG of complexation (from no interaction to full interaction) – ΔG of solvation (from no interaction to full interaction) + ΔG of going from no restraint to full restraint at 0 interaction lambda + ΔG of removing the restraint at full interaction energy.

The latest version of Tinker is available at <https://github.com/jayponder/tinker>. Tinker-OpenMM is available at <https://github.com/pren/tinker-openmm>. Note that Tinker only works using the modified Tinker-OpenMM, not the main OpenMM release.

Results

Force agreement

Correct simulation of molecular systems requires an accurate calculation of both force and energy. However, since energy is

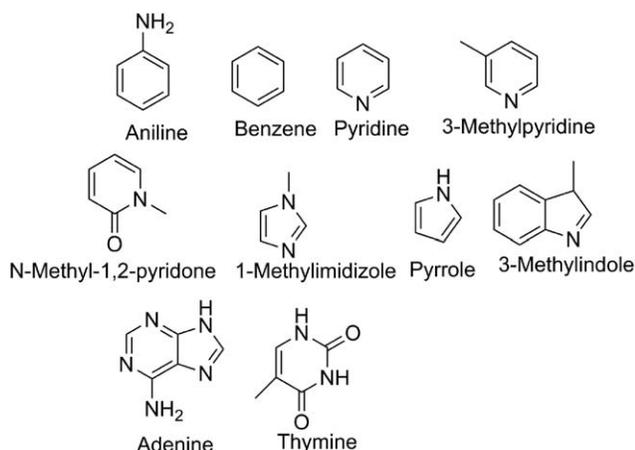


Figure 4. Structures of the 10 aromatic compounds used in this study.

only utilized by Tinker in the BAR process, and is not used during OpenMM molecular dynamics, we focused our initial analysis of Tinker-OpenMM on agreement of OpenMM forces with those of Tinker. To ensure that lambda was working in the Tinker-OpenMM implementation, we tested molecule 1 of the *sampl4* dataset bound to the host at a range of lambda values, and compared the resulting static forces to those of Tinker. The Tinker-OpenMM platform was able to closely match that of Tinker for all tested lambda values, with a root mean squared error of approximately 8.6×10^{-4} Kcal/mol/Å, and a maximal atomic force deviation of approximately 4.7×10^{-3} Kcal/mol/Å (Table 1). These degrees of deviation are negligible when considering that the RMS force is 31 Kcal/mol/Å. The force deviation is partially due to the single precision used in GPU force evaluation.

Computational efficiency

To test the speed and scalability of the Tinker-OpenMM platform, we ran 1000 steps of MD on *sampl4* molecule 1 (6417 atoms), and the *bench7* test case distributed with Tinker (a protein system of 23,558 atoms). For both test systems, the NVidia GTX1070 and GTX 970 were approximately 66-fold and 40-fold faster than an eight core CPU simulation, respectively (Table 2). A single CPU core is approximately 200-fold slower than simulation on a GTX1070 due to the poor core scalability of Tinker utilizing OpenMP. The GPU platform shows better than linear scaling with respect to system size, with a 3.7-fold increase in particle number resulting in a 2.4-fold or 2.5-fold

Table 1. Force comparison between the Tinker-AMOEBA CPU and Tinker-OpenMM-AMOEBA GPU platforms for *Sampl4* molecule 1 at a range of lambda values.

VDW lambda/ ele-lambda	RMSE force (10 ⁻⁴ Kcal/mol/Å)	Max force deviation (10 ⁻³ Kcal/mol/Å)
1/1	8.58	4.69
1/0.5	8.59	4.66
1/0.0	8.58	4.71
0.5/0.0	8.58	4.72
0.0/0.0	8.58	4.72

Table 2. Performance of Tinker-OpenMM on Nvidia GTX1070 and GTX970 GPUs without the relative binding calculations compared to Tinker CPU running on 8 OpenMP threads (4X of single CPU speed).

	GTX1070	GTX970	CPU
mol01(6417 atoms)	20.0	12.2	0.3
bench7(23558 atoms)	8.3	4.8	0.16

Values are in nanoseconds/day.

decrease in speed on the GTX1070 and GTX970 platforms, respectively. This better than linear scaling is likely a result of the smaller *sampl4* systems being unable to saturate GPU core utilization, as verified by profiling GPU core utilization during simulations. The change of the vdW force to the softcore 14–7 force resulted in no observable difference in speed compared to the kernel used in OpenMM. This was confirmed by running simulations using a version of Tinker-OpenMM that had been modified to utilize a standard, non-softcore 14–7 vdW force without the presence of the lambda parameter in the codebase.

To test the cost of utilization of relative vdW, tests were run on *bench7* with the relative VDW activated by using two waters (atoms 9000–9002 and 9003–9005) as “ligands” for the alchemical dual topology process. Both of these waters had their *ele-lambda* values set at 0.0, with a utilized *vdW-lambda* of 1.0. This allowed for the activation of dual topology kernels without introducing extra costs. This system was minimized, and a speed test was run as above. This resulted in a speed of 4.68 ns/day on a GTX 970, an approximately 2.5% speed reduction when compared to the absolute simulations. This small cost is only present when doing relative free energy

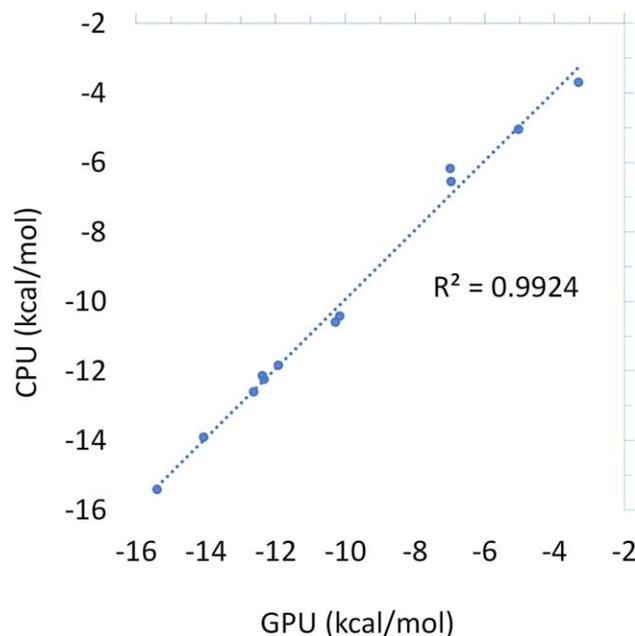


Figure 5. Comparison between the *sampl4* binding free energies of 12 *sampl4* compounds computed by the Tinker-OpenMM GPU and Tinker CPU platforms. GPU simulations were run for 5 ns at each perturbation step, while CPU simulations were run for 1 ns. [Color figure can be viewed at www.onlinelibrary.com]

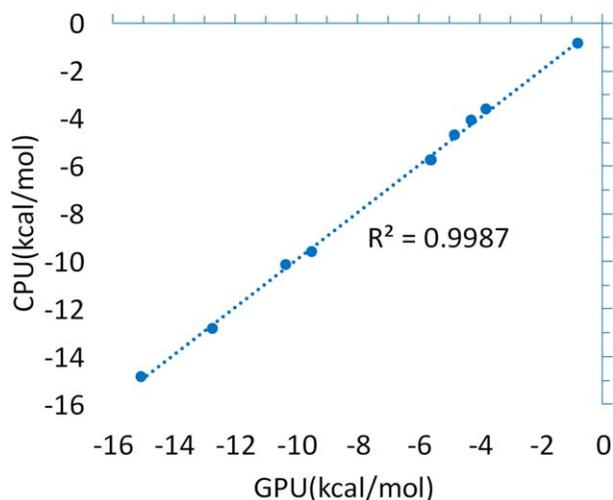


Figure 6. Comparison between the calculated solvation free energies for the 10 molecule aromatic compound dataset on the Tinker-OpenMM GPU and Tinker CPU platforms. [Color figure can be viewed at wileyonlinelibrary.com]

calculations; when no *ligand 2* parameter is set, the cheaper absolute vdW kernel is used for force and energy calculation.

Tinker-OpenMM defaults to a utilizing a “mixed” precision mode in all calculations. This mixed precision mode uses 32-bit floating point calculation for all forces, and integrates using 64-bit floating point precision. Due to the poor double floating point calculation of the consumer GeForce line of graphics cards, the use of double precision for both integration and force calculation results in an 18.1-fold reduction in performance on a GTX 970.

GPU/CPU absolute free energy agreement

As a test of the ability of the Tinker-OpenMM platform to reproduce the results of the Tinker CPU implementation, we performed hydration free energy calculation on a dataset of 10 aromatic compounds, as well as binding free energies on 12 ligands of the *sampl4* dataset (9). Both the solvation (Fig. 5) and *sampl4* datasets (Fig. 6) show agreement within the uncertainty of BAR, with R^2 values of (0.9924) and (0.9987), respectively. This, along with the static force calculations provides strong evidence that the GPU and CPU implementations of the AMOEBA force field produce comparable results. The fact that a high degree of agreement is possible despite the

Table 3. Comparison between the Tinker-OpenMM absolute and relative platform calculation of the solvation energy between pairs of aromatic compounds.

	Relative from Dual-Topology	Difference by Absolute
Aniline/Benzene	4.2 ± 0.1	4.0 ± 0.1
Adenine/Pyrrrole	11.4 ± 0.1	11.3 ± 0.1
Aniline/Adenine	-10.2 ± 0.1	-10.2 ± 0.1
Benzene/3-Methylimidazole	-9.0 ± 0.1	-8.7 ± 0.1
3-Methylpyridine/pyridine	-0.1 ± 0.1	0.0 ± 0.1

Values are in Kcal/mol.

Table 4. Comparison between the Tinker-OpenMM absolute and relative platform calculations of the relative binding free energy between pairs of *sampl4* compounds.

	mol05-mol06		mol09-mol10	
	Relative from absolute GPU	Relative from dual topology	Relative from absolute GPU	Relative from dual topology
Complexation energy	44.3 ± 0.1	44.3 ± 0.1	-56.3 ± 0.1	-56.0 ± 0.1
solvation energy	47.3 ± 0.1	47.3 ± 0.1	-68.0 ± 0.1	-68.0 ± 0.1
total $\Delta\Delta G$	-2.9 ± 0.1	-2.9 ± 0.1	10.4 ± 0.2	10.7 ± 0.1

Values are in Kcal/mol.

fact that the GPU simulations were run for 5 times longer (5 ns vs. 1ns at each perturbation step) is an indication that the tested systems converge relatively rapidly.

GPU/CPU relative free energy agreement

We then proceeded to test the capability of the dual-topology-based relative free energy platform by computing the relative solvation values for the aromatic dataset. For all tested aromatic pairs, the relative hydration free energy values computed from the dual-topology approach and the absolute HFE showed an agreement within 0.3 Kcal/mol, with an R^2 value of 0.999 (Table 3). The observed deviation is likely a result of random, nonsystematic statistical error.

Finally, we tested the relative binding prediction of two pairs of *sampl4* compounds. The first set of compounds, mol05 and mol06 share similar scaffolds, and show agreement in both complexation and solvation to within the uncertainty of BAR (Table 4).

The relative binding between molecules 9 and 10 constitutes a more challenging case that cannot be handled using the dummy atom-based approach due to the lack of a shared scaffold. In addition, this dissimilarity between the ligands may theoretically make convergence more difficult in the intermediate vdW transitions. Nonetheless, the relative binding platform was still able to agree with the absolute platform to within 0.3 Kcal/mol, demonstrating the advantage of dual-topology platform.

Discussion and Conclusions

This work reports a GPU implementation of alchemical free energy simulation for polarizable force field AMOEBA. The enhanced speed of GPU over CPU will be valuable for applications such as lead optimization. We have shown that the Tinker-OpenMM GPU platform is capable of reproducing the results of Tinker CPU platform, with an approximately 200-fold improvement in computational performance over what is possible on a single CPU core. This usage of GPU computation greatly improved sampling, which should allow for accounting for slow dynamics such as induced fit effects and other local changes in protein structure. Therefore, we expect the better

sampling afforded by the GPU-based platform will potentially lead to improved accuracy in ligand binding free energy prediction.

In addition to raw performance, one of the biggest challenges facing the free energy calculation field is the application of techniques to improve sampling of flexible systems to enable convergence with lesser simulation times. One methodology to achieve this increase in sampling efficiency is the calculation of relative binding free energies. Unlike previously utilized dummy atom-based approaches,^[78–82] the framework presented here is general and does not require a shared set of atoms to be utilized effectively. A special path has been designed to avoid unstable ligand–ligand polarization in the dual-topology approach. We expect that for flexible protein systems, the dual-topology approach will be more efficient and reduce the time needed for convergence in comparison with absolute free energy approaches.

Keywords: AMOEBA · free energy calculation · graphics processing units · Tinker · OpenMM

How to cite this article: M. Harger, D. Li, Z. Wang, K. Dalby, L. Lagardère, J.-P. Piquemal, J. Ponder, P. Ren. *J. Comput. Chem.* **2017**, DOI: 10.1002/jcc.24853



Additional Supporting Information may be found in the online version of this article.

- [1] E. Sanz, C. Vega, *J. Chem. Phys.* **2007**, *126*, 014507.
- [2] D. Seeliger, B. L. De Groot, *Biophys. J.* **2010**, *98*, 2309.
- [3] J. Aqvist, *J. Phys. Chem.* **1990**, *94*, 8021.
- [4] N. M. Garrido, A. J. Queimada, M. Jorge, E. A. Macedo, I. G. Economou, *J. Chem. Theory Comput.* **2009**, *5*, 2436.
- [5] S. Jayaraman, E. J. Maginn, *J. Chem. Phys.* **2007**, *127*, 214504.
- [6] J. D. Chodera, D. L. Mobley, M. R. Shirts, R. W. Dixon, K. Branson, V. S. Pande, *Curr. Opin. Struct. Biol.* **2011**, *21*, 150.
- [7] L. G. Ferreira, R. N. dos Santos, G. Oliva, A. D. Andricopulo, *Molecules* **2015**, *20*, 13384.
- [8] D. T. Moustakas, P. T. Lang, S. Pegg, E. Pettersen, I. D. Kuntz, N. Brooijmans, R. C. Rizzo, *J. Comput. Aided Mol. Des.* **2006**, *20*, 601.
- [9] R. A. Friesner, J. L. Banks, R. B. Murphy, T. A. Halgren, J. J. Klicic, D. T. Mainz, M. P. Repasky, E. H. Knoll, M. Shelley, J. K. Perry, D. E. Shaw, P. Francis, P. S. Shenkin, *J. Med. Chem.* **2004**, *47*, 1739.
- [10] M. L. Verdonk, J. C. Cole, M. J. Hartshorn, C. W. Murray, R. D. Taylor, *Proteins* **2003**, *52*, 609.
- [11] Y. Tanrikulu, B. Kruger, E. Proschak, *Drug Discov. Today* **2013**, *18*, 358.
- [12] G. L. Warren, C. W. Andrews, A. M. Capelli, B. Clarke, J. LaLonde, M. H. Lambert, M. Lindvall, N. Nevins, S. F. Semus, S. Senger, G. Tedesco, I. D. Wall, J. M. Woolven, C. E. Peishoff, M. S. Head, *J. Med. Chem.* **2006**, *49*, 5912.
- [13] M. Fischer, R. G. Coleman, J. S. Fraser, B. K. Shoichet, *Nat. Chem.* **2014**, *6*, 575.
- [14] I. J. Enyedy, W. J. Egan, *J. Comput. Aided Mol. Des.* **2008**, *22*, 161.
- [15] W. L. Jorgensen, *Science* **2004**, *303*, 1813.
- [16] D. Mobley, M. K. Gilson, *Annu. Rev. Biophys.* **2017**, *46*, 531.
- [17] M. K. Gilson, H.-X. Zhou, *Annu. Rev. Biophys. Biomol. Struct.* **2007**, *36*, 21.
- [18] W. L. Jorgensen, J. K. Buckner, S. Boudon, J. Tiradorives, *J. Chem. Phys.* **1988**, *89*, 3742.
- [19] M. K. Gilson, J. A. Given, B. L. Bush, J. A. McCammon, *Biophys. J.* **1997**, *72*, 1047.
- [20] B. Roux, *Comput. Phys. Commun.* **1995**, *91*, 275.
- [21] S. Izrailev, S. Stepaniants, B. Isralewitz, D. Kosztin, H. Lu, F. Molnar, W. Wriggers, K. Schulten, In *Computational Molecular Dynamics: Challenges, Methods, Ideas*; Springer: Berlin, **1999**; pp. 39–65.
- [22] J. R. Gullingsrud, R. Braun, K. Schulten, *J. Comput. Phys.* **1999**, *151*, 190.
- [23] G. M. Torrie, J. P. Valleau, *J. Comput. Phys.* **1977**, *23*, 187.
- [24] T. W. Allen, O. S. Andersen, B. Roux, *Biophys. Chem.* **2006**, *124*, 251.
- [25] D. Zhang, J. Gullingsrud, J. A. McCammon, *J. Am. Chem. Soc.* **2006**, *128*, 3019.
- [26] H.-J. Woo, B. Roux, *Proc. Natl. Acad. Sci. USA* **2005**, *102*, 6825.
- [27] A. Y. Lau, B. Roux, *Nat. Struct. Mol. Biol.* **2011**, *18*, 283.
- [28] S. Doudou, N. A. Burton, R. H. Henchman, *J. Chem. Theory Comput.* **2009**, *5*, 909.
- [29] T. P. Straatsma, H. J. C. Berendsen, *J. Chem. Phys.* **1988**, *89*, 5876.
- [30] A. Barducci, G. Bussi, M. Parrinello, *Phys. Rev. Lett.* **2008**, *100*, 020603.
- [31] A. Laio, F. L. Gervasio, *Rep. Prog. Phys.* **2008**, *71*, 126601.
- [32] G. Bussi, A. Laio, M. Parrinello, *Phys. Rev. Lett.* **2006**, *96*, 090601.
- [33] L. Zheng, M. Chen, W. Yang, *Proc. Natl. Acad. Sci. USA* **2008**, *105*, 20227.
- [34] L. Zheng, M. Chen, W. Yang, *J. Chem. Phys.* **2009**, *130*, 06B618.
- [35] C. H. Bennett, *J. Comput. Phys.* **1976**, *22*, 245.
- [36] M. R. Shirts, J. D. Chodera, *J. Chem. Phys.* **2008**, *129*, 129105.
- [37] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, M. Karplus, *J. Comput. Chem.* **1983**, *4*, 187.
- [38] K. Vanommeslaeghe, E. Hatcher, C. Acharya, S. Kundu, S. Zhong, J. Shim, E. Darian, O. Guvench, P. Lopes, I. Vorobyov, *J. Comput. Chem.* **2010**, *31*, 671.
- [39] J. B. Klauda, R. M. Venable, J. A. Freites, J. W. O'Connor, D. J. Tobias, C. Mondragon-Ramirez, I. Vorobyov, A. D. MacKerell, Jr., R. W. Pastor, *J. Phys. Chem. B* **2010**, *114*, 7830.
- [40] A. D. MacKerell, N. Banavali, N. Foloppe, *Biopolymers* **2000**, *56*, 257.
- [41] D. A. Pearlman, D. A. Case, J. W. Caldwell, W. S. Ross, T. E. Cheatham, S. DeBolt, D. Ferguson, G. Seibel, P. Kollman, *Comput. Phys. Commun.* **1995**, *91*, 1.
- [42] J. Wang, R. M. Wolf, J. W. Caldwell, P. A. Kollman, D. A. Case, *J. Comput. Chem.* **2004**, *25*, 1157.
- [43] A. Pérez, I. Marchán, D. Svozil, J. Sponer, T. E. Cheatham, C. A. Loughton, M. Orozco, *Biophys. J.* **2007**, *92*, 3817.
- [44] K. L. Meagher, L. T. Redman, H. A. Carlson, *J. Comput. Chem.* **2003**, *24*, 1016.
- [45] P. Y. Ren, J. W. Ponder, *J. Phys. Chem. B* **2003**, *107*, 5933.
- [46] J. C. Wu, G. Chatterjee, P. Ren, *Theor. Chem. Acc.* **2012**, *131*, 1138.
- [47] Y. Shi, Z. Xia, J. Zhang, R. Best, C. Wu, J. W. Ponder, P. Ren, *J. Chem. Theory Comput.* **2013**, *9*, 4046.
- [48] J. R. Maple, Y. Cao, W. Damm, T. A. Halgren, G. A. Kaminski, L. Y. Zhang, R. A. Friesner, *J. Chem. Theory Comput.* **2005**, *1*, 694.
- [49] H. A. Stern, G. A. Kaminski, J. L. Banks, R. Zhou, B. Berne, R. A. Friesner, *J. Phys. Chem. B* **1999**, *103*, 4730.
- [50] G. A. Kaminski, H. A. Stern, B. J. Berne, R. A. Friesner, *J. Phys. Chem. A* **2004**, *108*, 621.
- [51] S. Patel, C. L. Brooks, *J. Comput. Chem.* **2004**, *25*, 1.
- [52] S. Patel, A. D. Mackerell, C. L. Brooks, *J. Comput. Chem.* **2004**, *25*, 1504.
- [53] C. M. Baker, V. M. Anisimov, A. D. MacKerell, Jr., *J. Phys. Chem. B* **2010**, *115*, 580.
- [54] P. E. Lopes, G. Lamoureux, B. Roux, A. D. MacKerell, *J. Phys. Chem. B* **2007**, *111*, 2873.
- [55] C. M. Baker, P. E. Lopes, X. Zhu, B. Roux, A. D. MacKerell, Jr., *J. Chem. Theory Comput.* **2010**, *6*, 1181.
- [56] P. Y. Ren, C. J. Wu, J. W. Ponder, *J. Chem. Theory Comput.* **2011**, *7*, 3143.
- [57] Y. Shi, C. J. Wu, J. W. Ponder, P. Y. Ren, *J. Comput. Chem.* **2011**, *32*, 967.
- [58] J. R. Abella, S. Y. Cheng, Q. Wang, W. Yang, P. Ren, *J. Chem. Theory Comput.* **2014**, *10*, 2792.
- [59] M. J. Schnieders, J. Baltrusaitis, Y. Shi, G. Chatterjee, L. Zheng, W. Yang, P. Ren, *J. Chem. Theory Comput.* **2012**, *8*, 1721.
- [60] D. Jiao, C. King, A. Grossfield, T. A. Darden, P. Y. Ren, *J. Phys. Chem. B* **2006**, *110*, 18553.
- [61] J. C. Wu, J. P. Piquemal, R. Chaudret, P. Reinhardt, P. Y. Ren, *J. Chem. Theory Comput.* **2010**, *6*, 2059.
- [62] A. Grossfield, P. Ren, J. W. Ponder, *J. Am. Chem. Soc.* **2003**, *125*, 15671.
- [63] D. R. Bell, R. Qi, Z. F. Jing, J. Y. Xiang, C. Mejias, M. J. Schnieders, J. W. Ponder, P. Y. Ren, *Phys. Chem. Chem. Phys.* **2016**, *18*, 30261.

- [64] J. Zhang, Y. Shi, P. Ren, Polarizable Force Fields for Scoring Protein–Ligand Interactions Protein-Ligand Interactions, 1st ed.; Wiley-VCH Verlag GmbH & Co. KGaA: Weinheim, Germany. **2012**; pp. 99–120.
- [65] Y. Shi, C. Z. Zhu, S. F. Martin, P. Ren, *J. Phys. Chem. B* **2012**, *116*, 1716.
- [66] D. Jiao, J. Zhang, R. E. Duke, G. Li, M. J. Schnieders, P. Ren, *J. Comput. Chem.* **2009**, *30*, 1701.
- [67] D. Jiao, P. A. Golubkov, T. A. Darden, P. Ren, *Proc. Natl. Acad. Sci. USA* **2008**, *105*, 6290.
- [68] J. Zhang, W. Yang, J. P. Piquemal, P. Ren, *J. Chem. Theory Comput.* **2012**, *8*, 1314.
- [69] L. Dagum, R. Menon, *IEEE Comput. Sci. Eng.* **1998**, *5*, 9.
- [70] C. Narth, L. Lagardère, E. Polack, N. Gresh, Q. Wang, D. R. Bell, J. A. Rackers, J. W. Ponder, P. Y. Ren, J. P. Piquemal, *J. Comput. Chem.* **2016**, *37*, 494.
- [71] L. Lagardère, F. Lipparini, E. Polack, B. Stamm, E. Cancès, M. Schnieders, P. Ren, Y. Maday, J. P. Piquemal, *J. Chem. Theory Comput.* **2015**, *11*, 2589.
- [72] P. Eastman, M. S. Friedrichs, J. D. Chodera, R. J. Radmer, C. M. Bruns, J. P. Ku, K. A. Beauchamp, T. J. Lane, L. P. Wang, D. Shukla, T. Tye, M. Houston, T. Stich, C. Klein, M. R. Shirts, V. S. Pande, *J. Chem. Theory Comput.* **2013**, *9*, 461.
- [73] P. Eastman, J. Swails, J. D. Chodera, R. T. McGibbon, Y. Zhao, K. A. Beauchamp, L.-P. Wang, A. C. Simmonett, M. P. Harrigan, B. R. Brooks, *BioRxiv* **2016**, 091801.
- [74] M. Levitt, *J. Mol. Biol.* **1983**, *170*, 723.
- [75] V. Hornak, C. Simmerling, *J. Mol. Graph. Model.* **2004**, *22*, 405.
- [76] R. Salomon-Ferrer, D. A. Case, R. C. Walker, *Wires Comput. Mol. Sci.* **2013**, *3*, 198.
- [77] J. C. Phillips, J. E. Stone, K. Schultent, In International Conference on High Performance Computing, Networking, Storage and Analysis; IEEE Press: Piscataway, NJ. **2008**; pp. 444.
- [78] W. L. Jorgensen, T. B. Nguyen, *J. Comput. Chem.* **1993**, *14*, 195.
- [79] N. Ota, C. Stroupe, J. Ferreira-da-Silva, S. A. Shah, M. Mares-Guia, A. T. Brunger, *Proteins* **1999**, *37*, 641.
- [80] S. Miyamoto, P. A. Kollman, *Proteins* **1993**, *16*, 226.
- [81] M. R. Reddy, M. D. Erion, *J. Am. Chem. Soc.* **2001**, *123*, 6246.
- [82] M. R. Reddy, V. N. Viswanadhan, J. N. Weinstein, *Proc. Natl. Acad. Sci. USA* **1991**, *88*, 10287.
- [83] M. R. Reddy, M. D. Erion, *J. Am. Chem. Soc.* **2001**, *123*, 6246.
- [84] S. H. Fleischman, C. L. Brooks, *J. Chem. Phys.* **1987**, *87*, 3029.
- [85] J. W. Ponder, F. M. Richards, *J. Comput. Chem.* **1987**, *7*, 1016.
- [86] C. Zhang, D. R. Bell, M. Harger, P. Y. Ren, *J. Chem. Theory Comput.* **2017**, *13*, 666.
- [87] H. S. Muddana, A. T. Fenley, D. L. Mobley, M. K. Gilson, *J. Comput. Aided Mol. Des.* **2014**, *28*, 305.
- [88] B. Roux, M. Nina, R. Pomès, J. C. Smith, *Biophys. J.* **1996**, *71*, 670.

Received: 5 March 2017

Accepted: 6 May 2017

Published online on 00 Month 2017